PROGRAMMER'S REFERENCE CARD



DDP-24 CHARACTER CODES	**This code can only be punched when the computer is punching the tape un- der program control.
------------------------	---

OCTAL	CTAL TYPEWRITER PAPER TAPE							-	_		either or	ackspace key is depressed n-line or off-line, a stop code				ADDRESS				
CODE	L/C	U/C	8	7	6 F	4	S	3	3 2	1	receives	generated; if the typewriter the code (54 ₈) either on-	PSEUDO-OP	DESCRIPTION	SYMBOL IN LOC	SYMBOL	OCTAL	DECIMAL	COMPOUND	
00	ø	ь			(<u> </u> :	\perp	\perp	$\perp \perp$	line or o	off-line, the carriage will be		D2501111 11011	200	51111502	001712	DEGITIAL	COMIT COME	
01	2		\vdash	4	+	+	•	+	0	0	Dackspac	ued.	ABS	Absolute	1			Ì		
03	3		+	+	-	-	+:	\perp	0		 									
04	4	: @	П	7			•		>	0		/ USED OCP ADDRESS CODES STANDARD EQUIPMENT	BCI	Binary Coded Information	×					
06	● 6	√		#		_	•	6		0	FUR:			Block Ending Symbol	х	х	х	х	х	
10	8	>	H	_		0		Ť			00000	Enable both character channels (Input and Output)	BSS	Block Starting Symbol	х	Х	х	х	х	
11	9 #			1		0	•	İ	0	0	00001	Enable input word channel	CALL	Subroutine Linkage	×	X				
20	*	¢	H		0 0						00002	Enable output word channel	DEC	Decimal constant	×			×		
22	S			-	0 0	2	<u> </u>	-	0		01000	Punch stop code								
23 24	T U	=		_	0 0	<u> </u>	•	C		0	02000	Typewriter input select (keyboard enabled)	ENBI	ITC with bit 10 = 1	×	х	X	×	X	
25 26	w	%	H		0	+	 :			0	02010	Typewriter output select	END	End of program						
27 30	X	1		_	0 0	_		-	0	0		(keyboard inhibited)	ENDM	End of macro						
31	Z				0	С		İ	工	0	02070	Disconnect standard I/O devices	EQU	EQU Equals		×	х	×	x	
33 36	tab				0 0		•	C	0		02100	Paper tape reader select	INAM	INA with bit 10 = 1	×	×	х	X	×	
40 41	_ 		لب	0		2	•	_	_	0	02200	Paper tape punch select	INHI	ITC with bit 10 = 0	×	Х	х	x	x	
42	К			0	0	э <u> </u>	<u> </u>	-	0		J									
43	L M)		0	-	5		-1-		0	COMMONLY	COMMONLY USED SKS ADDRESS CODES FOR STANDARD EQUIPMENT		Produce Listing						
45 46	N 0	*Δ		0	-	+	+:		0 0	0	FOR			Macro definition	×	x	х	×		
47	Р	,		0		0	•	(0		00001	Sense switch #1	MOR	More tape						
50 51	Q R			0	+	0 0		-	+	0	00002	Sense switch #2	MZE	7					.,	
53	\$			0		5 0				0	00004	Sense switch #3		Minus Zero	X	Х	X	×	×	
54 56	backsp space	ace*	-	0	-	0 0		_	0 0	*	00010	Sense switch #4	NLST	Do Not Produce Listing		х				
60	& A	&	F	_	0 0	0 0	•	-	—	0	00020	Sense switch #5	NTRY	Subroutine Entry]	
62	B				0 0	0	1	+	0		00040	Sense switch #6	ост	Octal Constant	х		х			
63	D	(0	0		+:		>		00100	Parity error	ORG	Program Origin	х	х	x	×	х	
65 66	E F			0	0 0			0	0 0	0	00200	Improper divide indicator	OTAM	OTA with bit 10 = 1	×	×	×	×	x	
67	G H	<		0	_		_		0	0	00400	Overflow indicator	PZE	Plus Zero	x	x	×	×	×	
71				0	0 0	0 0		1	_	0	01000	Stop code							^	
73	lower	v shift	<u>. </u>		0 0) •		_	0	11000	Word output channel ready	REL	Relocatable						
75 76	upper car. re		_	0		C		0	0 0	0	12000	Word input channel ready	RTRN	Subroutine Exit	x	x			x	
77	line fe				0 10			-	0 0		14000	Character I/O channel	SWT	Sense Switch Test	х	x	Х	Х	х	
stop	backsp		0	_		ī				: 1	7	ready			"					

SUMMARY OF DAP PSEUDO-OPERATIONS

1. The assembly program (DAP) will recognize the & (60 $_3$) as a + 2. The assembly program (DAP) will recognize the line feed (77 $_6$) as a delete

OCTAL	P-CODE MNEMONIC	INSTRUCTION	EXECUTION TIME (µs)		OP-CO MNEMONIC	DE OCTAL	FUNCTION	X I O'F		OP- MNEMONIC	CODE OCTAL	FUNCTION	х	l 0'F	
00	HLT	Halt	5		CRA	60	0 → (A)			JMP	74	Jump to EA	Р	Р	
02	XEC	Execute	5+		IAB	57	(A) ++(B)			JOF	73	Jump to EA, if overflow indicator	Р	P	
03	STB	Store B	10	Æ	LDA	24	(EA) → (A)	PP		JPL	70	set; reset overflow indicator	_		
04	STC	Store Command Portion of A	10	STORE	LDB	23	(EA) →(B)	PP		JRT	70 25	Jump to EA, if sign of (A) is + Jump to location specified by (EA) ₁₀₋₂₄		P P	
. 05	STA	Store A	10	∞	STA	05	(A) → (EA)	PP	JUMP	JK1	25	restore interrupt	۲	۲	
06	STD	Store Address Portion of A	10	OAD	STB	03	(B) → (EA)	РР	3	JST	27	Jump to EA + 1 and store location	P	Р	
07	INM	Input to Memory	10	_	STC	04	(A) ₁₋₉ →(EA) ₁₋₉	PP				in (EA) ₁₀₋₂₄	_		
10	ADD	Add	10		STD	06	(A) ₁₀₋₂₄ + (EA) ₁₀₋₂₄	PP		JZE	71	Jump to EA, if (A) = ± 0		P	
11	SUB	Subtract	10		TAB	55	(A) → (B)			SKG	12	Skip next instruction, if (A) > (EA)		P	
12	SKG	Skip if A Greater	10-12 10-12							SKN	13	Skip next instruction, if (A) ≠ (EA)	Р	Р	
13	SKN	Skip if A Not Equal	10-12	ı	400	10		PPP		ADX**	54	$(X) + (EA)_{10-24} \rightarrow (X)$	R	P	
15 16	ANA ORA	AND to A OR to A	10		ADD ADM	10 20	$(A) + (EA) \rightarrow (A)$ $(A) + I(EA)I \rightarrow (A)$	PPP		IRX	67	$(EA)_{10-24} + 1 \rightarrow (EA)_{10-24}$ and (X) ,	P	P	
17	ERA	Exclusive OR to A	10		BCD*	36	1	PP	×			See Manual			
20	ADM	Add Magnitude	10		BIN*	37	(EA) BCD → (A) Binary (EA) Binary → (B) BCD	PPP	INDEX	JIX	72	Jump to EA, if $(X) \neq 0$		P	
21	SBM	Subtract Magnitude	10	윤	DIV	35	(A, B) / (EA) → (quotient to B,		=	JXI	75	$(X) + 1 \rightarrow (X)$, Jump to EA, if resultant $(X) \neq 0$	R	P	
22	OTM	Output from Memory	10	ARITHMETIC] "	,,,	remainder to A)			LDX**	56	(EA) ₁₀₋₂₄ → (X)	R	P	
23	LDB	Load B	10	Ę	MPY	34	(B) X (EA) → (A, B)	P P		STX	66	$(X) \rightarrow (EA)_{10-24}$		P	
24	LDA	Load A	10	₹	RND	62	$(A) + 1 \rightarrow (A)$, if $(B)_2 = 1$	P		TAX	63	$(A)_{10-24} \to (X)$	R		
25	JRT	Jump Return	10		SBM	21	(A) - I(EA)I→ (A)	PPP							
27	JST	Jump and Store Location	10		SMP	30	See Manual	PPP		DMB	32	Dump memory starting at EA, See Manual	R	P	
30	SMP	Step Multiple Precision	10		SUB	11	(A) - (EA) → (A)	PPP		FMB	31	Load memory starting at EA,	R	p	
31	FMB	Fill Memory Block	variable		ļ					,		See Manual	.,		
32	DMB	Dump Memory Block	variable		}				-	INA**	52	Input → (A), according to mask	Р	Р	
34	MPY	Multiply	31	٩٢	ANA	15	(A) AND (EA) →(A)	P P	D G	1444	0.7	in (EA), See Manual		_	
35	DIV	Divide	33	LOGICAL	ERA	17	(A) Exclusive OR (EA) → (A) (A) OR (EA) → (A)	PP	Ę,	INM ITC**	07 51	Input → (EA) Inhibit or enable interrupt, accord-		P P	
36*	BCD*	BCD to Binary Conversion*	33	2	ORA	16		PP	NPUT/OUTPUT	110~	21	ing to mask in (EA)	Р	Р	
37*	BIN*	Binary to BCD Conversion*	33		1				Ā	OCP**	53	Select 1/0, according to mask	Р	Р	
40	ARS	A Right Shift	5+n	-	 							in (EA)			
41	ALS	A Left Shift	5+n 5+n 5+n		ALS**	41	Shift (A) ₂₋₂₄ left, positions	P P		OTA**	50	(A) → Output, according to mask in (EA), See Manual	Р	P	
42 43	LRR LLR	Long Right Rotate Long Left Rotate					specified by (EA) ₁₉₋₂₄			отм	22	(EA) → Output	P	Р	
44	LRS	Long Right Shift	5+n		ARS**	40	Shift (A) ₂₋₂₄ right, positions specified by (EA) ₁₉₋₂₄	PP							
45	LLS	Long Left Shift	5+n		LGL**	47	Shift (A) ₁₋₂₄ left, positions	PP		SKS**	61	Skip next instruction, if sense line	Р	P	
46	NRM	Normalize	5+n				specified by (EA) ₁₉₋₂₄			1		not set; sense line specified by (EA)			
47	LGL	Logical Left Shift	5+n		LLR**	43	Rotate (A, B) ₁₋₂₄ left, positions specified by (EA) ₁₉₋₂₄	PP	ROL	ľ					
50	OTA	Output from A	5		LLS**	45	i e	P P	CONTROL	HLT	00	Stop computer operation until start button pressed			
51	ITC	Interrupt Control	5		LLS	43	Shift (A, B) ₂₋₂₄ left, positions specified by (EA) ₁₉₋₂₄		ű	NOP	77	Perform no operation			
52	INA	Input to A	5		LRR*	42	Rotate (A.B) 24 right, positions	PP		XEC	02	Execute instruction at EA	Р	Р	
53	0CP	Output Control Pulse	5		1.0044	4.0	specified by (EA) ₁₉₋₂₄			l					
54	ADX	Add to Index	5	SHIFT	LRS**	44	Shift (A, B) ₂₋₂₄ right, positions specified by (EA) ₁₉₋₂₄	PP		**If indirect a	address not sp	ecified ($I=0$), address portion of instruction	n Is		
55	TAB	Transfer A to B	5	₹	NRM	46	Shift (A, B) ₂₋₂₄ left until (A) ₂ =1	P		effective of	perand.				
56	LDX	Load Index			1		See Manual								
57	IAB	Interchange A and B	10 5		SCL**	65	Shift (A, B) ₂₋₂₄ left and decrement index register, positions	R P				WORD FORMATS			
60	CRA	Clear A					specified by (EA) ₁₉₋₂₄			WORD FORMAL					
61 62	SKS RND	Skip if Sense Line Not Set Round A	5 6		SCR**	SCR** 64	Shift (A, B) ₂₋₂₄ right and increment	R P	18	STRUCTION WO	pn.	DATA WORD			
63	TAX	Transfer A to Index	5				index register, positions specified by (EA) ₁₉₋₂₄		F	TO TRUCTION WO	VO.	DATA WURD			
64	SCR	Scale Right	!	5+n											
65	SCL	Scale Left	5+n		X = Indexa	ble									
66	STX	Store Index	10		I = Indirec	tly Addressab	ple	Ļ	1 3.4 _ 9.10 24						
67	IRX	Increment, Replace, and Load Index	14	į	0'F = Overflo	ow possible				144 4	ADDRE		TUDE		
70	JPL	Jump if A Plus	6		* Option										
71	JZE	Jump if A Zero	5		** If indi	rect address n	ot specified ($I = 0$), address portion of inst	ruction				W 0005			
72	JIX	Jump on Index	5		is effe			OPERATION CODE SIGN							

--- INDEX EXPANSION

- INDIRECT ADDRESS

- INDEX

77 *OPTIONAL

73

74

75

J0F

JMP

JXI

NOP

No Operation

Jump on Overflow

Unconditional Jump

Jump on Index Incremented

5

5

7

5

P = Possible

R = Required

(P) = Improper divide possible